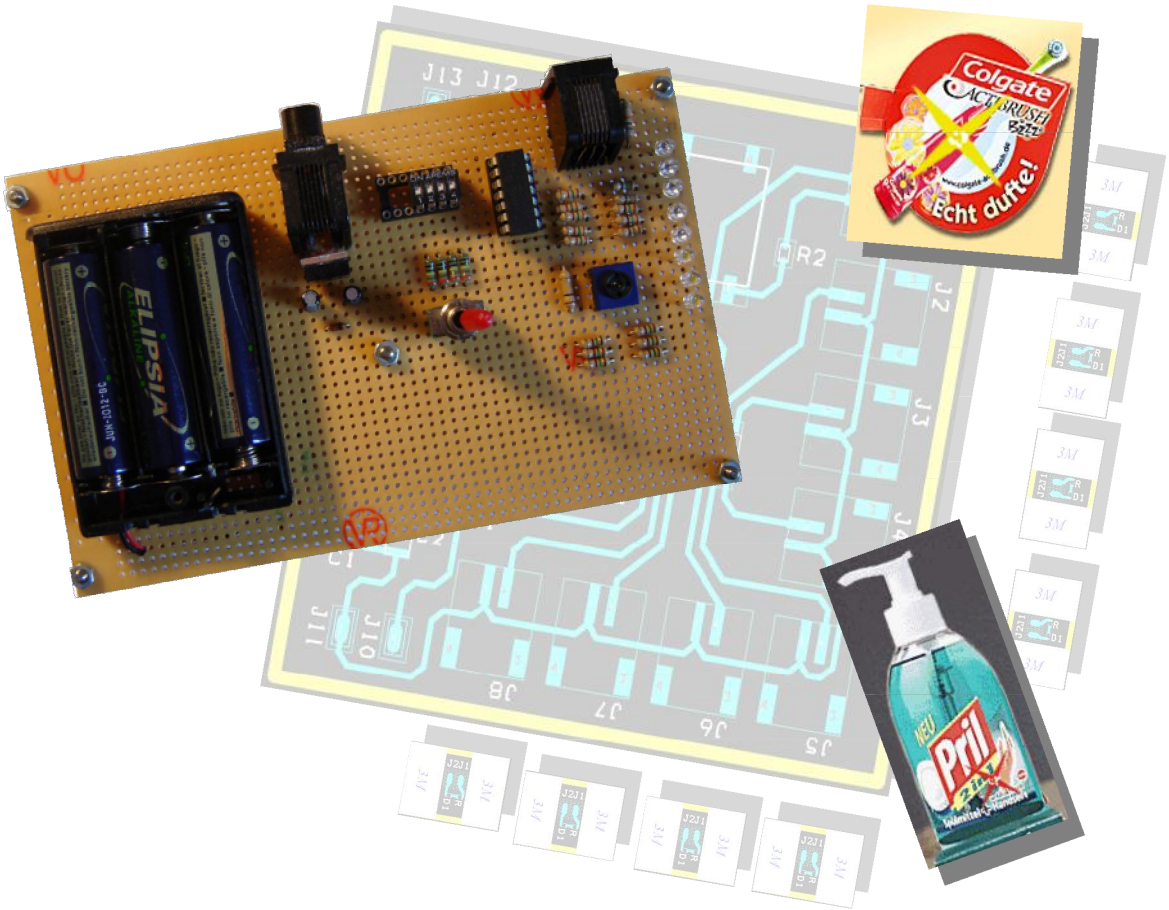
Projet industriel 4ème année

DEVELOPPEMENT D'UN SYSTEME DE CONTRÔLE DE LED A APPLICATION PLV



Tuteurs entreprise : M. HOUEL
M. LABIDURIE

Tuteurs ENIVL : M. COLAMARTINO
M. TRAN

Etudiants de 4ème année : Emilie COURBOULEIX
Michaël TURPIN

Juin 2007

Remerciements

Nous remercions l'entreprise TRAFF de nous avoir accueillis dans le cadre de notre projet industriel de 4ème année.

Nous remercions également nos tuteurs d'entreprise, M. Houel et M. Labidurie, ainsi que nos tuteurs de l'ENIVL, M. Colamartino et M. Tran, de nous avoir épaulés pour mener à bien ce projet.

Résumé

Nous avons choisi d'effectuer notre projet industriel de 4ème année dans l'entreprise TRAFF, spécialisée dans la conception de systèmes lumineux à LED, afin d'acquérir des compétences dans ce domaine.

Nous avons eu en charge le développement d'un système de LED à application PLV (Promotion sur le Lieu de Vente). Ce genre de système permet une publicité animée sur un produit et ainsi attire davantage l'attention du client.

Après s'être renseignés sur le type de produit proposé par la concurrence, nous avons pu en dégager notre cahier des charges pour faire face au marché. Ensuite, a suivi une étude de détermination et dimensionnement des composants destinés au prototype. Après programmation du microcontrôleur, la maquette a pu être testée dans les futures conditions d'utilisation.

Enfin le dossier de réalisation, destiné à la fabrication à grande échelle, a été monté, regroupant la liste complète des composants, les fichiers de routage réalisés avec le logiciel Orcad et des précisons à propos du montage. Un manuel d'utilisation en français et en anglais a également été rédigé.

Mots clés : animation, carte de contrôle, CMS, LED, microcontrôleur, PIC, PLV, programmation, routage, temporisation.

Sommaire

Remerciements……

Résumé ……1

Sommaire ……2

Introduction ……3

1. Présentation du projet……4
1.1. L'entreprise ……4
1.2. Veille technologique ……4
1.3. Le cahier des charges……6

2. Réalisation du prototype……7
2.1. Compléments au cahier des charges……7
2.2. Dimensionnement des composants……7
2.3. Conception du prototype ……8

3. Programmation ……12
3.1. Configuration du microcontrôleur……12
3.2. Structure du programme ……14
3.3. Description des animations……16
3.4. Calcul des temporisations ……17

4. Finalisation du produit……21
4.1. Type de composants ……21
4.2. Elaboration du dossier de réalisation ……22
4.3. Rédaction du manuel d'utilisation ……25

Conclusion……26

Sources documentaires ……27

Introduction

A l'heure du développement durable, la réduction de la consommation d'énergie est un enjeu pour toutes les sociétés. Les LED ne nécessitent que très peu de courant (aux alentours de 20 mA) et grâce aux dernières avancées technologiques, elles procurent aujourd'hui une forte intensité lumineuse. De plus en plus utilisées dans le monde de l'industrie (automobile, signalétique, marketing,…), nous avons choisi de réaliser notre projet industriel chez TRAFF, spécialiste des applications lumineuses industrielles à LED.

Le gérant de TRAFF, Ludovic Labidurie, nous a demandé de nous attaquer à la niche commerciale des applications PLV où l'entreprise Loupi est numéro un. Dans le but d'être compétitif, il est impératif d'être moins cher ou alors à prix égal mais avec des caractéristiques techniques plus performantes.

Afin de mener à bien ce projet, nous avons établi un cahier des charges basé sur le principal produit concurrent en y ajoutant quelques améliorations ne portant pas atteinte au coût de revient. Ensuite, un prototype a été développé pour tester et valider les spécifications du cahier des charges. Enfin, un dossier de réalisation et un guide d'utilisation ont été rédigés pour finaliser le développement du produit.

1. Présentation du projet

1.1. L'entreprise

Forts de plus de 10 ans d'expérience dans le développement et la commercialisation de feux à LED (diode électroluminescente), Ludovic LABIDURIE (gérant) et Damien GROLLEAU créent fin janvier 2004 la SARL TRAFF (pour traffic).

Abandonnant la première idée d'investir le marché des feux à LED, l'idée leur est venue d'occuper une « niche », c'est-à-dire une activité en expansion à faible pression concurrentielle. L'entreprise est spécialisée dans les applications d'éclairage industriel à LED principalement pour renouveler l'offre des systèmes d'éclairages proposée par les « enseignistes » aux commerces et entreprises.

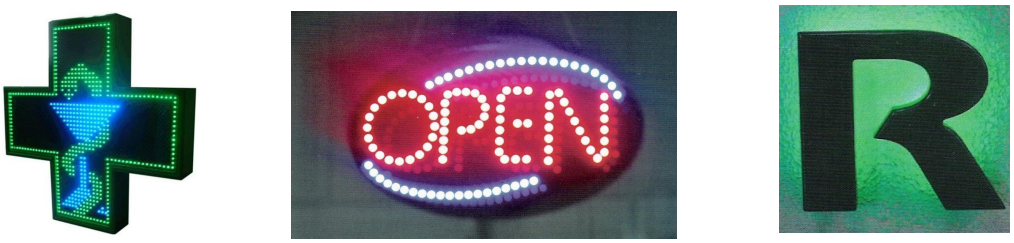


Fig. 1 : Exemples d'applications proposées par TRAFF

L'entreprise s'organise autour de deux plateformes professionnelles :

- Le siège social, situé à COUR-CHEVERNY (41- FRANCE), se compose d'un bureau d'étude, d'une équipe marketing/commerce et de la logistique, regroupant ainsi un effectif de six employés.
- La seconde est située à Shanghai et comprend le bureau des méthodes, la production et le contrôle qualité.

Ce type d'organisation, offrant souplesse et réactivité, leur permet d'appréhender avec la même rigueur tous types de développement spécifique.

Pour répondre à une demande d'un de leurs clients, mais aussi pour concurrencer la société Loupi, monopolisateur sur ce genre de produit, TRAFF nous a confié le développement d'un kit PLV (Promotion sur le Lieu de Vente).

1.2. Veille technologique

Nous nous sommes intéressés aux produits proposés par la société Loupi, dans le but d'offrir un produit ayant les mêmes fonctionnalités à un prix moindre ou plus complet au même prix.

Loupi & Electronic Art Gallery est n°1 de l'électronique appliquée à la publicité en proposant d'intégrer ses systèmes d'animation dans les campagnes de PLV ou de marketing direct. Elle se situe à Paris et Düsseldorf, le stock produits en Allemagne et le stock composants en Chine.

Parmi toute la gamme de produits (montage 1 LED, à alimentation solaire, badge lumineux,…), nous nous sommes centrés sur ceux équipés de microcontrôleur.

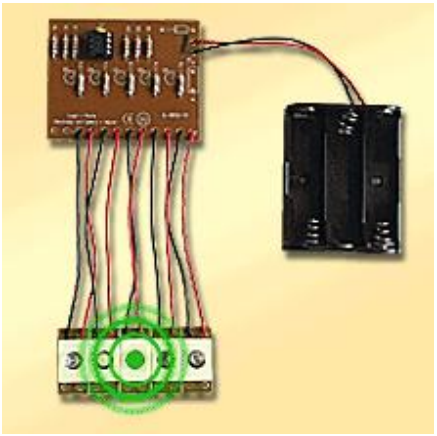


Fig. 2 : Circuit 5 LED

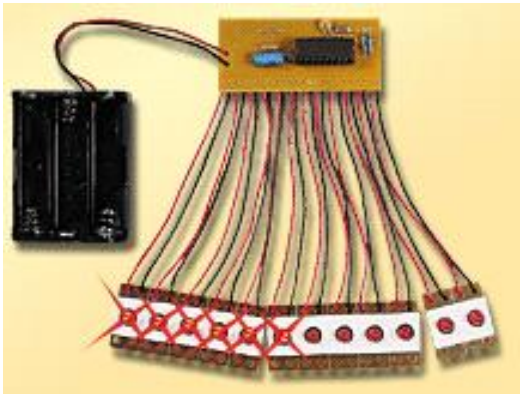


Fig. 3 : Circuit 12 LED

CARACTERISTIQUES	
5 LED Ø 5 mm transparentes	12 LED Ø 5 mm diffusantes
1 animation selon le cahier des charges	1 animation selon le cahier des charges
Autonomie non spécifiée	1 mois non stop avec 3 piles alcalines
Circuit de commande environ 62x50x10 mm	Circuit de commande environ 65x30x10 mm

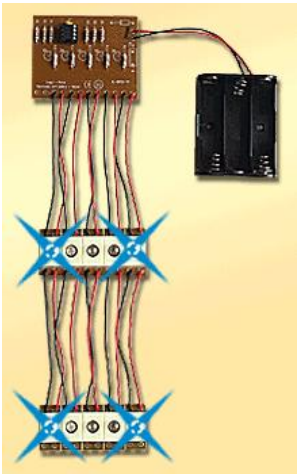


Fig. 4 : Circuit 10 LED

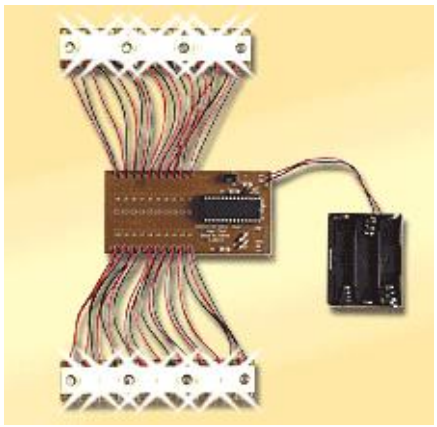


Fig. 5 : Circuit 20 LED

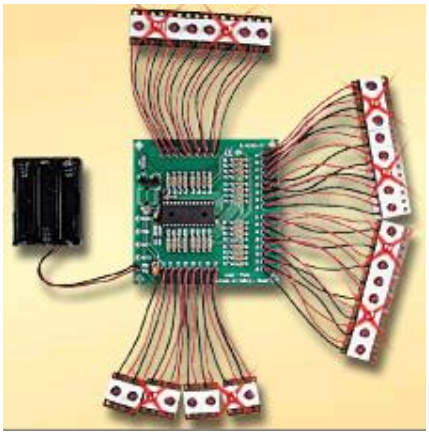


Fig. 6 : Circuit 32 LED

CARACTERISTIQUES		
5x2 LED Ø 5 mm diffusantes	20 LED Ø 5 mm diffusantes	32 LED Ø 5 mm diffusantes
1 animation selon le cahier des charges	1 animation selon le cahier des charges	1 animation selon le cahier des charges
Autonomie non spécifiée	Autonomie non spécifiée	+ ou – 1 mois non stop avec 3 piles alcalines.
Dimension du circuit non spécifiée	Circuit de commande environ 105x53x10 mm	Circuit de commande environ 100x100x10 mm

Aucun prix n'est fourni sur le site internet de Loupi car les produits sont adaptés aux désirs du client et réservés aux professionnels. Il faut donc les contacter par mail ou par téléphone. Nous n'avons réussi à obtenir que le prix indicatif de 8€/unité pour un lot de 200 cartes à une LED d'autonomie d'un mois.

1.3. Le cahier des charges

Voici ci-dessous le cahier des charges, défini avec l'entreprise, sur lequel nous nous sommes appuyés pour la conception du prototype.

- *Objectifs*

  Développer un kit PLV de taille minimale qui permet de promouvoir un produit se trouvant à l'intérieur d'un magasin à l'aide d'animations lumineuses à LED pendant une durée de 4 semaines.

Fig. 7 : Exemple de kit PLV déjà existant

- *Description des fonctionnalités*

  → 1 entrée d'alimentation 5 V.
  → 1 entrée pour la batterie.
  → 5 sorties pour les LED (20 mA/sortie).
  → Autonomie de 4 semaines minimum.
  → Animations : chenillard, fondu, flash (3 flashs consécutifs puis temps mort), clignotement (temps d'allumage est égal au temps éteint),…
  → Etre le moins cher possible.

- *Questions soulevées*

  → Faut-il, par souci d'un coût trop élevé, faire une carte par type d'animation ou une carte pouvant effectuer plusieurs animations ?
  → Le choix d'un DIP Switch est-il judicieux pour sélectionner les animations ?

2. Réalisation du prototype

Avant de choisir les composants pour la conception de la maquette de test, il est primordial de trouver les réponses aux questions soulevées par le cahier des charges et de préciser ou dimensionner le choix des composants.

2.1. Compléments au cahier des charges

- *Estimation du surcoût qu'engendre le fait de pouvoir sélectionner différentes animations sur une même carte*

Les composants supplémentaires, pour permettre le choix de l'animation, ont été répertoriés et leurs prix ont été relevés sur les sites de fournisseurs de composants électroniques. Le tableau de comparaison suivant a pu être établi :

Une animation par carte		*Plusieurs animations par carte*	
Composants	**Prix**	**Composants**	**Prix**
PIC12F635	$0,80*	PIC16F636	$0,94*
		DIP Switch (4 entrées)	≈ $0,53**
		4 résistances	négligeable
	$0,80		**≈ $1,47**

* prix d'achat en volume affiché sur le site de Microchip.

**tarifs affichés sur le site de Radiospares : 42 switches à 17€ (0,40€/switch).

Le surcoût sera donc d'environ **0,50 €** par carte. Vu cette faible valeur, TRAFF nous a dit d'opter pour une carte à plusieurs animations.

- *Solution technologique pour la sélection des animations*

En ce qui concerne le moyen de sélection des animations, la première solution envisagée était celle avec le DIP Switch. L'utilisateur devait changer les positions des interrupteurs, quatre au total, pour choisir l'animation. Mais très vite une alternative s'est présentée en l'utilisation d'un bouton poussoir. En effet, à ce moment là, la tâche est largement facilitée puisqu'il suffit de presser le bouton pour changer d'animation. Ceci rend possible d'augmenter le nombre de LED car des ports du microcontrôleur sont libérés (8 LED au lieu de 5).
Les deux solutions citées ont été réalisées sur le prototype, mais seule celle avec le bouton poussoir, plus fonctionnelle, sera gardée pour le produit final.

2.2. Dimensionnement des composants

- *Choix du type et du nombre de piles de la batterie*

Le microcontrôleur peut être alimenté par une tension comprise entre 2,0 et 5,5 V.

Lorsque l'alimentation externe est utilisée, il est souhaité que le microcontrôleur soit alimenté en 5 V continu. Pour cela un régulateur 5 V est utilisé pouvant supporter jusqu'à 34 V continu en entrée.
Si l'alimentation par pile est choisie, des piles alcalines de 1,5 V seront utilisées car leur capacité (nombre d'ampère que peut fournir une pile pendant une heure) est assez élevée pour ce genre de produit (2500mAh). Deux piles auraient pu suffire (2x1,5=3,0 V), mais au fur et à mesure de leur utilisation, les piles délivrent de moins en moins de tension et la limite d'alimentation du PIC de 2,0 V serait vite atteinte. C'est pour cette raison qu'il est préférable de mettre 3 piles.

- *Résistance de protection*

Pour qu'une LED émette de la lumière, il faut qu'elle soit alimentée par un courant suffisant (I_F). Si le courant est de faible intensité ($I<I_F$), la LED éclaire faiblement, si le courant est de forte intensité, la LED est détruite et à un courant légèrement élevé, sa durée de vie est beaucoup plus courte. Il faut donc déterminer la valeur de la résistance de protection.

Caractéristiques de la LED : I_F = 20 mA, V_F = 1,7 V

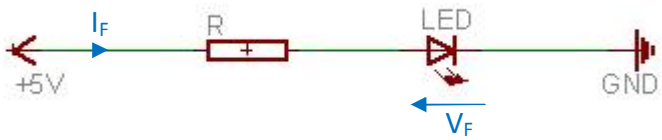


Fig. 8 : Schéma de la résistance de protection et de la LED

Le schéma du dessus donne la relation pour trouver R :

$$R = \frac{V_{PIC} - V_F}{I_F} = \frac{5 - 1{,}7}{0{,}02} = 165\,\Omega$$

2.3. Conception du prototype

D'après les propriétés des microcontrôleurs fabriqués par Microchip, le PIC 16F636 est le plus approprié à nos besoins (PIC 8 bits, 12 sorties, mémoire flash, RAM 126 Mo,…). Malheureusement ce PIC n'étant pas disponible sur le site de Radiospares, le PIC de référence 16F684 lui sera substitué (seulement pour le prototype), le plus proche du point de vue des propriétés.

Tous les composants nécessaires ont été listés et commandés sur le site internet de Radiospares pour obtenir rapidement les composants sans chercher les prix les plus bas.
Le tableau suivant détaille les composants pour chaque type de sélection d'animations. Ceux en vert sont commun aux deux solutions, ceux en jaune correspondent au DIP Switch et ceux en bleu au bouton poussoir.

Nom du composant	Quantité	Repères sur les schémas
PIC 16F684	1	PIC 16F684
LED (3mm, 20 mA)	5	LED1, LED2, LED3, LED4, LED5
Résistance 150 Ω	5	R1, R3, R5, R7, R9
Résistance 15 Ω	5	R2, R4, R6, R8, R10
Connecteur Jack (alimentation)	1	<=+5V
Régulateur 5 V (LM7805CV)	1	LM7805CV
Condensateurs 0,33 µF	2	C1, C2
Support à piles (3x1,5 V)	1	3x1,5 V
Diode	1	D1
Connecteur RJ11 (dialogue carte/PC)	1	RJ11
Résistance 10 Ω	1	R15
Interrupteur	1	
Carte à trous	1	
DIP Switch APEM 4 contacts	1	DIP switch
Résistance 1 MΩ	4	R11, R12, R13, R14
Bouton poussoir	1	BOUTON_POUSSOIR
Résistance 100 kΩ	1	R22
LED (3mm, 20 mA)	3	LED6, LED7, LED8
Résistance 150 Ω	3	R16, R18, R20
Résistance 15 Ω	3	R17, R19, R21

Fig. 9 : Tableau des composants du prototype

Une fois les composants reçus, il ne reste plus qu'à les souder ou les wrapper sur la carte selon les schémas électroniques que nous avons réalisés à l'aide du logiciel Eagle (cf. Fig. 10 et Fig. 11 page suivante). Nous avons élaboré un schéma pour chaque solution car avec un unique schéma trop de fils se croisaient ce qui le rendait difficilement lisible. Un interrupteur permet de basculer en mode DIP Switch ou bouton poussoir.

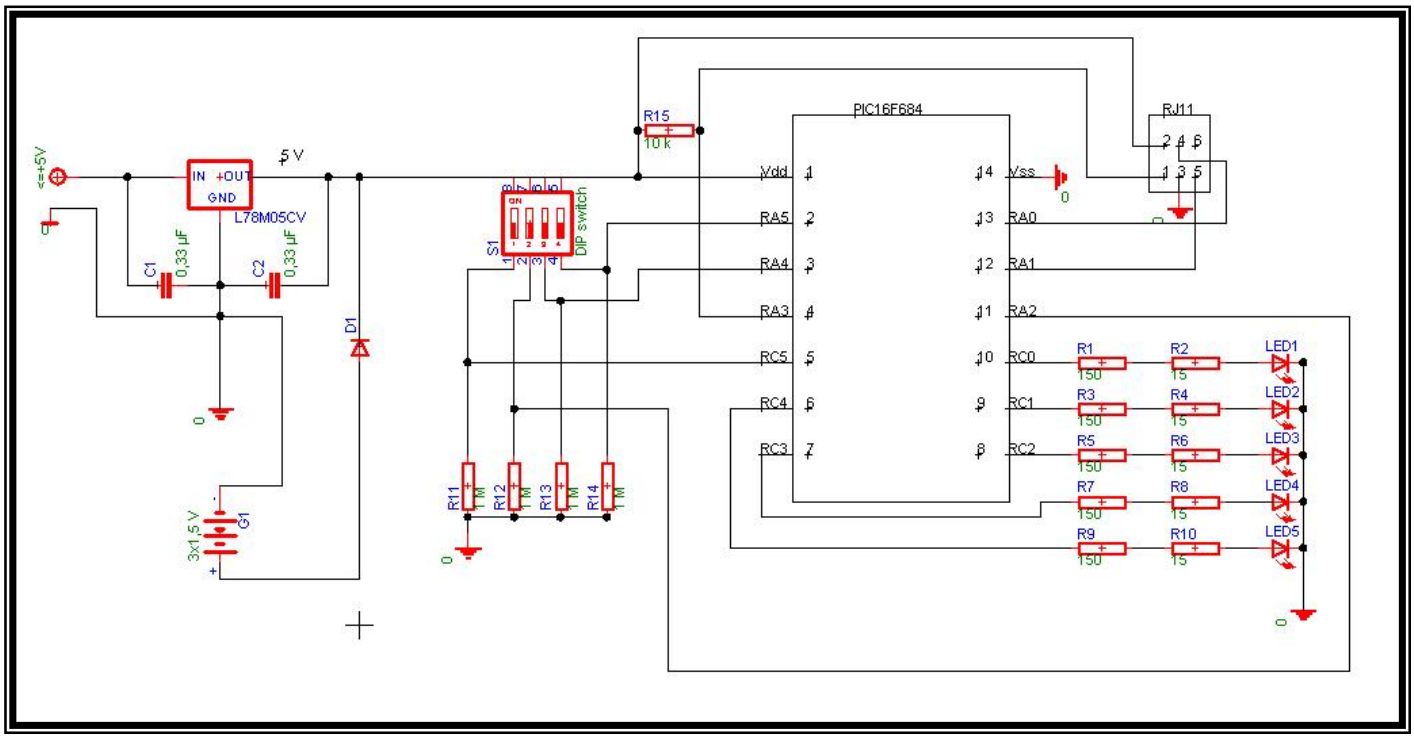


Fig. 10 : Schéma du circuit avec le DIP Switch

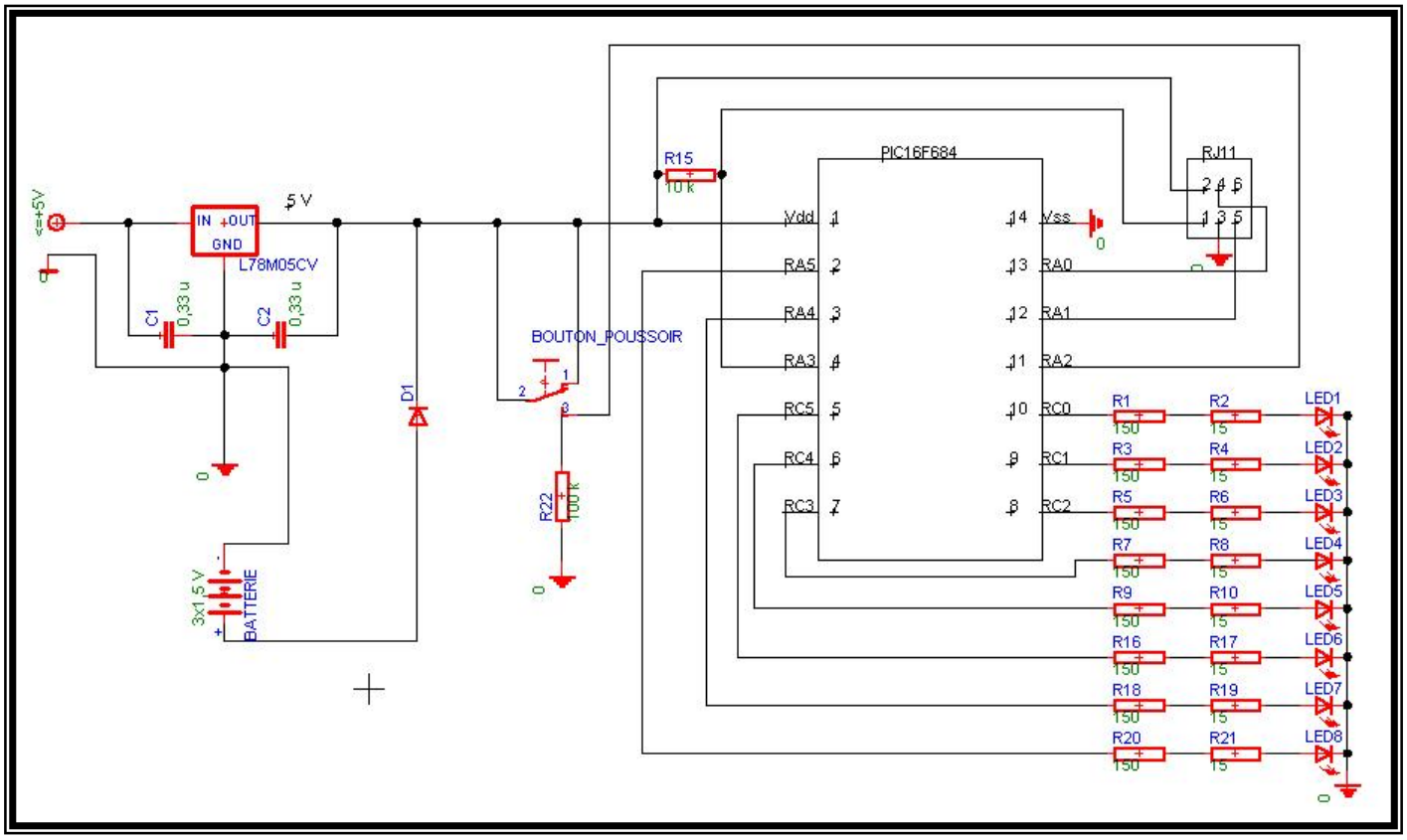


Fig. 11 : Schéma du circuit avec le bouton poussoir

A part le microcontrôleur et le DIP Switch qui sont montés sur des supports de wrapping, tous les autres composants sont soudés directement à la carte. Les liaisons entre le DIP Switch et microcontrôleur ainsi que celles entre le microcontrôleur et les LED ont été wrappées.

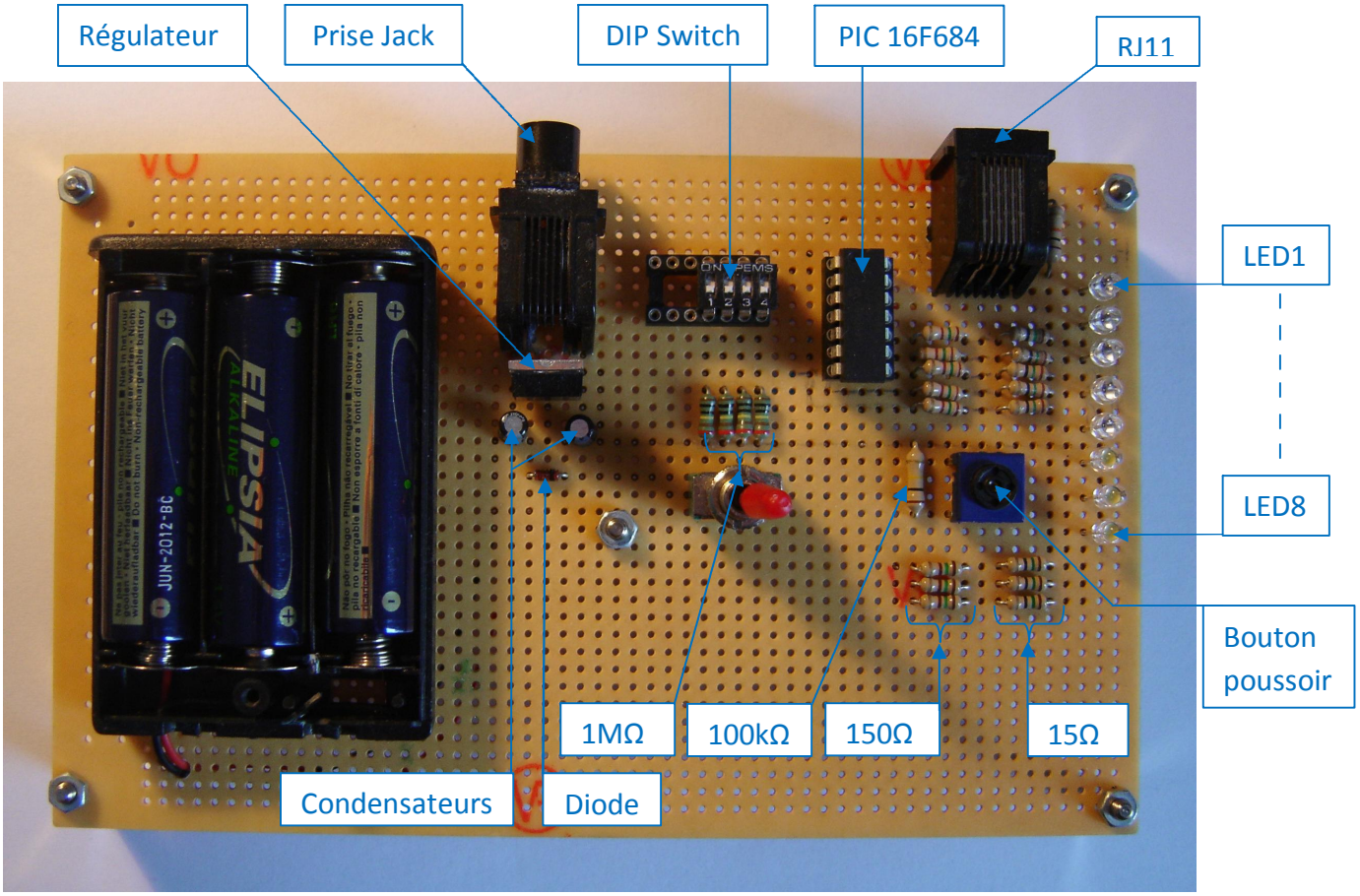


Fig. 12 : Vue de dessus du prototype

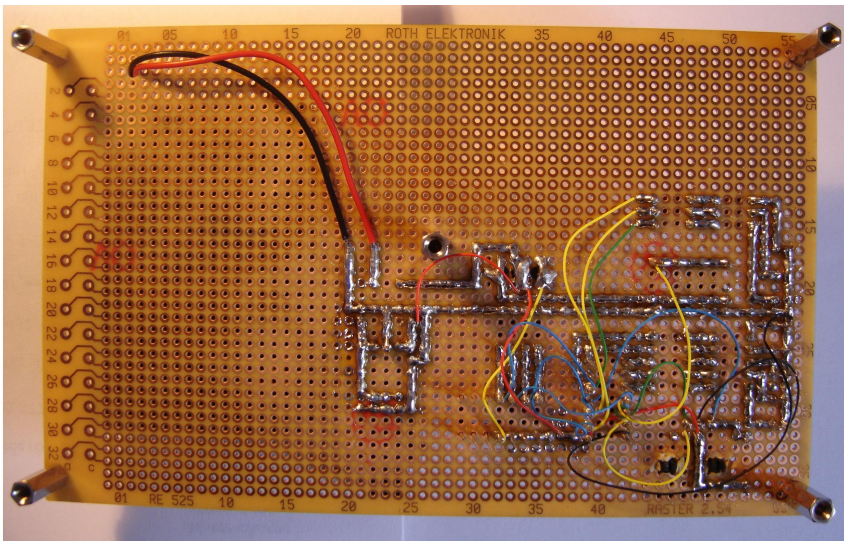


Fig. 13 : Vue de dessous du prototype

Maintenant il ne reste plus qu'à entreprendre la programmation du microcontrôleur.

3. Programmation

Nous avions commencé par programmer le PIC pour l'utilisation du DIP Switch, mais une fois qu'il a été décidé que le bouton poussoir serait la solution gardée, nous n'avons pas réalisé toutes les animations pour le DIP Switch (6 sur les 10). Dans cette partie, seul le programme développé pour le bouton poussoir sera explicité.

3.1. Configuration du microcontrôleur

Avant de commencer à réfléchir à la structure même du programme, il faut tout d'abord initialiser correctement les ports et les registres du microcontrôleur de façon à ce qu'il réponde à la fonction qu'il doit remplir. Cette étape est fondamentale car si elle est mal effectuée, malgré un programme sans erreur, nous n'arriverons pas à obtenir le résultat escompté.

- *Configuration des ports*

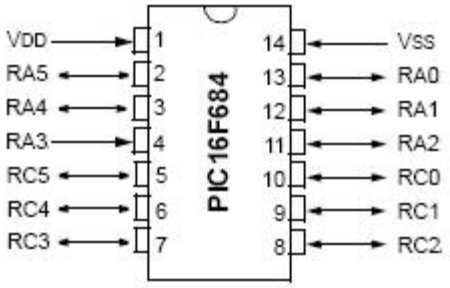


Fig. 14 : Schéma des pins

Le PIC 16F684 ne possède en tout que deux ports : le port A et le port C qui peuvent être configurés en entrée ou en sortie (sauf pour le pin RA3 qui est toujours en entrée). Il est alimenté par la tension d'entrée appliquée au pin V_{DD} et relié à la masse par V_{SS}. La communication avec l'ordinateur par le RJ11 est réservée aux pins RA0, RA1, et RA3 (configurés en entrée). Les LED 1 à 8 sont branchées aux pins RC0, RC1, RC2, RC3, RC4, RC5, RA4 et RA5 (configurés en sortie) et le bouton poussoir sera sur le pin RA2 (configuré en entrée). Maintenant que le rôle de chaque pin est déterminé, les valeurs à charger sur les ports A et C peuvent être définies (1 en entrée, 0 en sortie).

PORTA

-	-	RA5	RA4	RA3	RA2	RA1	RA0
0	0	0	0	1	1	1	1

Valeur à charger : $0F

PORTC

-	-	RC5	RC4	RC3	RC2	RC1	RC0
0	0	0	0	0	0	0	0

Valeur à charger : $00

- *Configuration du Timer 1*

Le Timer 1 sera utilisé pour réaliser les temporisations. C'est un compteur 16 bits allant de 0 à 65 535. Lorsqu'il est activé, il s'incrémente tous les cycles machine. Dans notre cas, un cycle machine vaut 1 µs, car l'horloge interne du microcontrôleur est à 4 MHz, cette valeur est ensuite divisée par 4, d'où la valeur de 1 µs (1/1 MHz). Lorsque le Timer 1 a fini de compter, il passe de 65535 à 0 en mettant à 1 le bit 0 nommé TMR1IF dans le registre PIR1.
Pour activer le Timer 1, il suffit de donner une valeur au registre T1CON.

T1CON

T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
0	0	0	0	0	0	0	1

→ T1GINV = 0, inutilisé
→ TMR1GE = 0, comme TMR1ON=1 Timer 1 activé
→ **T1CKPS1 et T1CKPS0 = 00, prédiviseur à 1**
→ T1OSCEN = 0, oscillateur interne désactivé
→ T1SYNC = 0, pas d'horloge externe
→ TMR1CS = 0, horloge interne de micro utilisée par le Timer 1
→ **TMR1ON = 1, Timer 1 activé**

Valeur à charger : $01

- *Activation des interruptions*

Une interruption est un événement spécifique qui interrompt le cours normal de l'exécution du programme pour traiter un événement et reprendre le programme principal à l'endroit où il avait été interrompu. Nous voulons utiliser ce principe pour que lorsque l'utilisateur appuie sur le bouton poussoir (relié au pin RA2) le programme enregistre le souhait de changer d'animation puis reprenne son fonctionnement normal. Il faut donc configurer les registres OPTION_REG et INTCON.

OPTION_REG

RAPU	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0
0	1	0	0	0	0	0	0

→ RAPU = 0, résistances de rappel du PORTA désactivées
→ **INTEDG = 1, interruption de RA2 sur front montant**
→ TOCS = 0, horloge interne du microcontrôleur utilisée par le Timer 0
→ TOSE = 0, comme TOCS = 0 bit sans fonction spéciale
→ PSA = 0, prédiviseur appliqué au Timer 0
→ PS<2 :0>= 000, prédiviseur = 2 (incrémentation du TMR0 tous les 2 événements)

Valeur à charger : $40

INTCON

GIE	PEIE	TOIE	INTE	RAIE	TOIF	INTF	RAIF
1	0	0	1	0	0	0	0

→ **GIE = 1, autorise les interruptions**
→ PEIE = 0, interruption de fin d'écriture en EEPROM inactive
→ TOIE = 0, interruption du Timer 0 inactive
→ **INTE = 1, interruption de RA2 active**
→ RAIE = 0, interruption sur le PORTA inactive
→ TOIF = 0, flag du Timer 0 initialisé à 0
→ **INTF = 0, flag de RA2 initialisé à 0**
→ RAIF = 0, flag de l'interruption sur la PORTA initialisé à 0

Valeur à charger : $90

Cette étape d'initialisation du PIC est fournie en annexe 1.

3.2. Structure du programme

Pour un programme comme celui-ci, le programme principal contient peu d'instructions. Comme on peut le voir sur l'organigramme Fig. 15, après avoir mis à zéro la mémoire du microcontrôleur (vecteur de reset) et fait toutes les initialisations dont celles décrites dans le paragraphe 3.1., le cœur du programme se résume donc à une boucle infinie faisant appel au sous-programme SelectAnim qui comme son nom l'indique permet de sélectionner l'animation à effectuer.

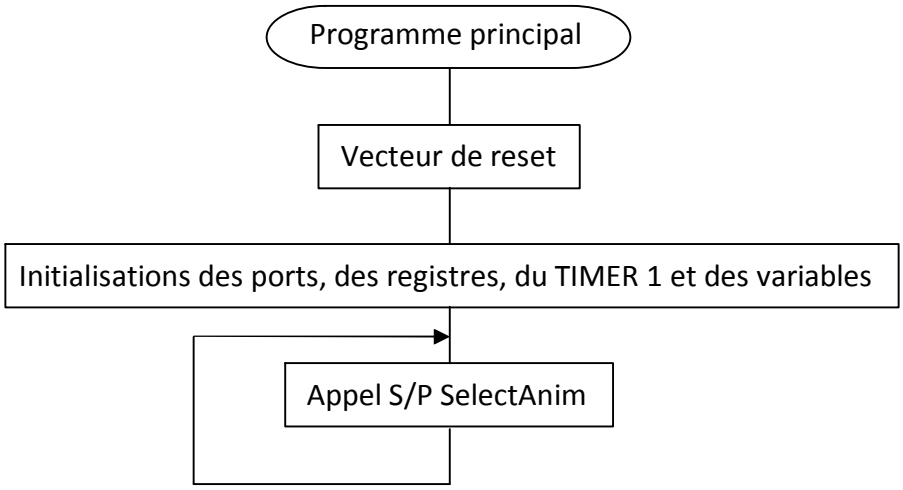

Programme principal
Vecteur de reset
Initialisations des ports, des registres, du TIMER 1 et des variables
Appel S/P SelectAnim

Fig. 15 : Organigramme du programme principal

Le plus gros du travail est rempli par ce fameux sous-programme SelectAnim. Son principe est basé sur le fait que l'on incrémente la variable *posbtn* à chaque fois que l'on appuie sur le bouton poussoir pour changer d'animation. A chaque valeur de cette variable correspond une animation : 0 pour le chenillard, 1 pour le k2000, 2 pour le flash,…

Fig. 16 : Organigramme du sous-programme SelectAnim

Il faut désormais définir précisément les différentes animations.

3.3. Description des animations

Les animations sont exécutées dans l'ordre suivant.

- *Animation « Chenillard »*

Chaque LED s'allume et s'éteint chacune à son tour. La première LED s'allume (les autres sont éteintes), puis la seconde et ainsi de suite.
Ordre d'allumage des LED : 1, 2, 3, 4, 5, 6, 7, 8 / 1, 2, 3, 4, 5, 6, 7, 8 / 1, 2,…

- *Animation « k2000 »*

Chaque LED s'allume l'une après l'autre dans un sens puis dans le sens inverse.
Ordre d'allumage des LED : 1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2 / 1, 2, 3, 4,…

- *Animation « flash »*

Toutes les LED s'allument trois fois successivement et ensuite s'éteignent toutes pendant un temps mort.

- *Animation « décalage intérieur »*

Les LED des extrémités s'allument, puis se décalent vers l'intérieur.
Ordre d'allumage des LED : 1&8, 2&7, 3&6, 4&5 / 1&8, 2&7, 3&6, 4&5 / 1&8, 2&7, 3&6,…

- *Animation « décalage extérieur »*

Les LED du milieu s'allument, puis se décalent vers celles des extrémités.
Ordre d'allumage des LED : 4&5, 3&6, 2&7, 1&8 / 4&5, 3&6, 2&7, 1&8 / 4&5,…

- *Animation « double décalage »*

Elle combine deux « chenillards » qui commencent à chaque extrémité pour se croiser au milieu.
Ordre d'allumage des LED : 1&8, 2&7, 3&6, 4&5, 3&6, 2&7 / 1&8, 2&7, 3&6,…

- *Animation « aléatoire »*

Toutes les LED s'allument aléatoirement.
Ordre d'allumage des LED : 4, 6, 8, 1, 3, 5, 7&2 / 4, 6, 8, 1, 3, 5, 7&2 / 4, 6, 8,…

- *Animation « fondu »*

Toutes les LED s'allument progressivement et s'éteignent pendant un temps mort.

- *Animation « clignotement »*

Toutes les LED clignotent régulièrement.

- *Animation « mixe »*

Les LED s'allument les unes après les autres dans un sens puis suivent 3 flashs, ensuite elles s'allument les unes après les autres dans le sens inverse et à nouveau 3 flashs.
Ordre d'allumage des LED : 1, 2, 3, 4, 5, 6, 7, 8, 3 flashs, 8, 7, 6, 5, 4, 3, 2, 1, 3 flashs / 1, 2, 3, 4, 5, 6, 7, 8, 3 flashs, 8, 7, 6, 5, 4, 3,…

3.4. Calcul des temporisations

Le cahier des charges nous a imposé une autonomie d'alimentation de 4 semaines, mais il faut considérer un coefficient de sécurité de 1,5 soit 6 semaines (1008 h). Il va donc falloir calculer le temps d'allumage et les temps morts pour chaque animation pour limiter au maximum la consommation.

Pour déterminer le temps d'allumage minimum d'une LED tout en voulant que son intensité lumineuse soit maximale, nous avons testé différentes valeurs de temporisation. Le temps de 10 ms permet à la LED d'émettre une lumière de forte intensité en un cours laps de temps. En dessous de cette valeur, la LED n'éclaire pas à son maximum.
Nous utiliserons alors une temporisation de 10 ms (**T_{all} = 10 ms**) pour l'allumage des LED dans toutes les animations.
Pour réaliser une temporisation de 10 ms, il faut attribuer une valeur aux variables TMR1H et TMR1L qui stockent la valeur sur 16 bits à partir de laquelle le Timer 1 compte. Si on veut être rigoureux, il faut prendre en compte les 5 µs que mettent les cinq premières instructions du sous-programme de temporisation. Le temps à compter sera alors 9995 µs. Il faut donc charger dans TMR1H et TMR1L la valeur suivante 65 535 – 9 995 = 55 540 ($D8F4, TMR1H = $D8 et TMR1L = $F4).

```
;************************
tempo1                  ; Temporisation de 10 ms pour allumer LEDs
    bcf     PIR1,0      ; Flag du TIMER 1 à 0
    movlw   0xF4        ; Charge 65535-9995=55540=$D8F4
    movwf   TMR1L       ; TMR1L=$F4
    movlw   0xD8        ; TMR1H=$D8
    movwf   TMR1H
lp1 btfss   PIR1,0      ; Si flag=1 on saute instruction suivante et tempo finie
    goto    lp1         ; Si flag=0 on boucle pour retester le flag
    return
;************************
```

Fig. 17 : Sous-programme de temporisation de 10 ms

Pour les temporisations des temps morts, nous ne chercherons pas à être aussi précis car même si la temporisation où la LED est éteinte dure quelques microsecondes de plus, ceci nous permet d'avoir une marge de sécurité pour répondre à l'autonomie imposée par le cahier des charges.

Tout d'abord il faut calculer le temps durant lequel une LED peut être allumée dans une minute. Nous savons déjà que la capacité moyenne des piles alcalines (LR6 ou AA) est de 2500 mAh. On en déduit les valeurs suivantes.

- Courant constant dans le circuit sur les 6 semaines :

  $\frac{2500\ mAh}{1008\ h}$ = 2,48 mA

- Consommation du montage :

  8 LED × 20 mA + 300 µA (PIC) = 160,3 mA

- Courant constant dans les LED sur les 6 semaines :
  2,48 - 0,3 = 2,18 mA
- Temps d'allumage d'une LED par minute :
  $\frac{2{,}18\ mA}{160\ mA}$ = 1,36% soit T_{LED} = 0,81 s/min

Maintenant il ne reste plus qu'à trouver les temps morts pour chaque animation.

- *Animations « Chenillard » et « k2000 »*

Le terme de « temps d'un cycle » représente le temps que met l'animation à allumer et éteindre chaque LED l'une après l'autre une seule fois (1, 2, 3, 4, 5, 6, 7, 8).

→ Temps d'un cycle : $T_c = 8\ (T_{all} + T_m)$

→ Nombre de cycles par minute : $\frac{810\ ms}{10\ ms}$ = 81 fois

→ Temps d'un cycle : $T_c = \frac{60\ 000\ ms}{81}$ = 740,7 ms

→ On en déduit alors que $T_{all} + T_m = \frac{740{,}7\ ms}{8}$ = 92,6 ms

→ Temps mort entre chaque allumage : 92,6 - 10 = 82,6 → T_m = 83 ms

- *Animation « flash »*

Ici deux temps morts sont à déterminer, un pour celui entre chaque flash (T_{m1}) et l'autre après les 3 flashs (T_{m2}).
Le terme de « temps d'un cycle » représente le temps que met l'animation à effectuer les 3 flashs successifs et le « long » temps mort.

→ Temps d'un cycle : $T_c = 3\ (T_{all} + T_{m1}) + T_{m2}$

→ Nombre de cycles par minute : $\frac{810\ ms}{30\ ms}$ = 27 fois

→ Temps d'un cycle : $T_c = \frac{60\ 000\ ms}{27}$ = 2 222,2 ms

→ On prend T_{m2} = 1,5 s comme long temps mort.

→ D'où le temps entre chaque flash

$T_{m1} = \frac{2\,222{,}2 - 1\,500 - 30}{3}$ = 230 ms → T_{m1} = 230 ms

- *Animations « décalage intérieur », « décalage extérieur » et « double décalage »*

Le terme de « temps d'un cycle » représente le temps que met l'animation à allumer toutes les LED une seule fois (exemple : 1&8, 2&7, 3&6, 4&5)

→ Temps d'un cycle : $T_c = 4\ (T_{all} + T_m)$

→ Nombre de cycles dans une minute : $\frac{810\ ms}{10\ ms}$ = 81 fois

→ Temps du cycle optimum dans une minute : $T_c = \frac{60\ 000\ ms}{81}$ = 740,7 ms

→ $T_m = \frac{740{,}7 - 40}{4}$ = 175 ms → T_m = 175 m

- *Animation « aléatoire »*

Le terme de « temps d'un cycle » représente le temps que met l'animation à allumer toutes les LED une seule fois (4, 6, 8, 1, 3, 5, 7&2).

→ Temps d'un cycle : $T_c = 7\ (T_{all} + T_m)$

→ Nombre de cycles dans une minute : $\frac{810\ ms}{10\ ms}$ = 81 fois

→ Temps du cycle optimum dans une minute :

$$T_c = \frac{60\ 000\ ms}{81} = 740{,}7\ ms$$

→ On en déduit alors $T_{all} + T_m = \frac{740{,}7\ ms}{7}$ = 105,8 ms

→ Temps mort entre chaque allumage :

T_m = 105,8 - 10 = 95,8 → T_m = 96 ms

- *Animation « fondu »*

Dans ce cas là, la temporisation d'allumage de 10 ms est trop rapide pour apercevoir la variation d'intensité lumineuse donc il faut trouver T_{all} et T_m.

→ On choisit d'allumer 17 fois une LED dans la minute, donc le temps d'allumage de la LED doit être égal à $T_{all} = \frac{0{,}81\ s}{17}$ = 47,6 ms → T_{all} = 47 ms

→ On en déduit le temps entre chaque fondu :

$$T_m = \frac{60 - (17 \times 0{,}0476)}{17} = 3{,}48\ s$$ → T_m = 3,5 s

- *Animation « clignotement »*

Le terme de « temps d'un cycle » représente le temps que met l'animation à allumer et éteindre toutes les LED.

→ Temps d'un cycle : $T_c = T_{all} + T_m$

→ Nombre de cycles par minute : $\frac{810\ ms}{10\ ms}$ = 81 fois

→ Temps du cycle : $\frac{60\ 000\ ms}{81}$ = 740,7 ms

→ Temps mort entre chaque allumage : 740,7 - 10 = 730,4 → T_m = 731 ms

- *Animation « mixe »*

Comme cette animation reprend les animations « k2000 » et « flash » elle répondra largement au cahier des charges.

Tableau récapitulatif des temporisations

ANIMATION	TEMPORISATION
« chenillard » « k2000 »	83 ms
« flash »	230 ms ; 1,5 s
« décalage intérieur » « décalage extérieur » « double décalage »	175 ms
« aléatoire »	96 ms
« fondu »	47 ms ; 3,5 s
« clignotement »	731 ms
« mix »	aucune

Pour limiter le nombre de lignes de code, seules les temporisations dites de base surlignées en magenta, seront codées et les autres seront réalisées en appelant plusieurs fois celles de base.

Tous les éléments sont regroupés pour coder les sous-programmes d'animations dont les organigrammes ainsi que les lignes de codes commentées sont fournis en annexes 2 à 15. Pour vérifier la validité des temporisations, la durée de vie du prototype a été testée. Equipé de 3 piles alcalines, le système s'est arrêté de fonctionner au bout de 6 semaines exactement. Bien entendu un affaiblissement de la luminosité des LED a pu être observé au cours du temps.
A ce stade, il ne reste plus qu'à chercher des fournisseurs pour les composants définitifs, établir le routage du circuit sur logiciel et rédiger le guide de l'utilisateur pour que le produit soit prêt à être commercialisé.

4. Finalisation du produit

4.1. Type de composants

Le Kit PLV doit être le plus petit possible pour pouvoir être dissimulé derrière le produit que l'utilisateur souhaite mettre en valeur. Aussi, il est préférable de pouvoir automatiser la production des cartes pour ainsi diminuer le coût global. C'est dans ces conditions que nous avons décidé d'utiliser la technologie CMS.
Le CMS (Composants Montés en Surface) est plus répandu sous son homologue anglais SMT (Surface-Mount Technology) et les composants sont appelés SMD (Surface-Mount Devices). Cette technologie consiste à souder les composants d'une carte à sa surface, plutôt que d'en faire passer les broches au travers. Cela permet entre autres de souder des composants des deux côtés de la carte électronique, d'avoir des composants de taille minimale (ex : 0,6x0,3 mm et même plus petit) et d'automatiser la production. Les CMS sont jusqu'à dix fois plus petits que leurs équivalents traditionnels, et leur coût peut être inférieur de 25 à 50 %.
Les composants CMS sont pourvus de petites terminaisons métalliques ou de petites broches à leurs extrémités pour pouvoir être brasés directement à la surface des circuits imprimés. Il existe alors deux techniques de brasage que nous allons succinctement décrire.

- *Brasage par refusion*

La refusion est utilisée pour les cartes contenant soit uniquement des composants CMS, soit des composants CMS dont le brasage ne peut se faire que par cette technique (BGA). La pose de composants CMS sur une carte est très simple et fiable par rapport aux composants traversants. Le circuit imprimé nu est d'abord sérigraphié : enduit à travers un pochoir métallique d'une pâte à braser sur les plages du circuit imprimé et les terminaisons des composants. Puis les composants sont posés sur le circuit par les machines de placement. Finalement le circuit passe dans un four, où la chaleur fait fondre la pâte déposée pour former la brasure.

- *Brasage à la vague*

Cette méthode est utilisée en cas de mixité des composants CMS et traditionnels. Elle consiste à déposer un point de colle aux futurs emplacements des composants au lieu de sérigraphier les plages. Les composants sont ensuite posés de la même manière que précédemment avant de polymériser la colle au four ou en étuve. Cela permet ensuite de placer des composants traditionnels (traversant le circuit imprimé) de l'autre côté de la carte.
La soudure se fait ensuite grâce à une vague d'étain en fusion, la carte passant au-dessus. Au contact de l'étain et par capillarité, les terminaisons des composants CMS et les broches des composants traversants sont soudées sur le circuit.

C'est ainsi une manière de mixer les deux technologies, l'intérêt est double :

- Tous les composants n'existent pas en version CMS.
- Certains doivent avoir une résistance mécanique supérieure à ce que la technologie CMS peut apporter (cas de certains connecteurs pour des questions de résistance à l'arrachement par exemple).

L'unique inconvénient concerne la maintenance du produit posant des problèmes supplémentaires aux techniciens assurant le dépannage, particulièrement lorsqu'ils doivent changer un composant.

En tout connaissance de cause, pour notre produit, afin toujours de minimiser les coûts de production, seuls des composants CMS (SMD) seront apposés sur une unique face de la carte électronique. Il s'agira donc d'un brasage par refusion pour fixer les composants et les pistes seront gravées sur une unique couche (le TOP=dessus).
Afin de pallier les problèmes de maintenance, les composants non utiles à l'utilisateur, ne servant qu'à la reprogrammation du PIC, ne seront pas fournis. Seules les pistes et les pastilles de contact seront gravées. Le technicien devra donc souder lui-même le connecteur RJ11 ainsi que la prise Jack pour utiliser une alimentation externe. En effet, lors d'un test pratique, l'alimentation par piles alcalines n'a pas permis de charger le programme dans le PIC. Il faut donc garder la possibilité d'avoir une alimentation externe régulée à +5V.
Enfin, nous choisirons des composants de taille moyenne car plus le composant est petit, plus il est cher.

Ainsi, le dossier de réalisation a été rédigé, décrivant la fabrication des cartes répondant à toutes les contraintes et nécessités, sera envoyé au fournisseur de l'entreprise TRAFF, situé en Chine. Dans ce pays, les coûts de main d'œuvre et de matière première optimisent le coût de production. Nous ne connaissons pas le coût de production mais il sera très faible, cela permettra de vendre le produit aux alentours de 8€ pièce et ne sera disponible que pour des grosses quantités (≥ 200).

4.2. Elaboration du dossier de réalisation

Nous ne décrirons ici que l'organisation et les principales idées contenues dans ce dossier car il est fourni en annexe 16.
Le dossier doit être clair, concis et précis. C'est dans cette optique que nous avons établi un plan chronologique dans la manière de fabriquer le produit. Tout d'abord le produit est présenté et son utilisation est décrite avec les principaux éléments. Ensuite, la liste des composants est donnée. Il s'agit de ceux que nous avons pu trouver chez divers fournisseurs sur internet à titre indicatif. En effet, le fabricant chinois possède ses propres fournisseurs et prendra les composants homologues.

Nous précisons aussi toute la connectique à avoir car nous avons décidé de permettre au client de choisir la longueur de câble qu'il souhaite pour connecter les huit cartes à LED à la carte de contrôle parmi une gamme de longueurs établie arbitrairement mais avec logique allant de 25 cm à 250 cm. Nous avons pris soin de respecter les couleurs selon qu'il s'agisse de la masse ou du $+V_{cc}$ (+4,5V avec les piles et +5V avec l'alimentation externe).
Ensuite, les fichiers sources pour le microcontrôleur sont fournis sur un CD et il ne reste plus qu'à les charger dans le PIC 16F636 avant même de l'implanter sur la carte.
Sur le CD joint au rapport sont aussi présents les fichiers GERBER permettant le routage automatisé des cartes à l'aide d'une machine appropriée. Afin d'obtenir ces fichiers, nous avons dû utiliser un logiciel spécifique, Orcad. D'autres existent tels que Eagle ou Proteus, mais Orcad est utilisé dans sa version d'essais à l'école d'où notre choix.
Tout d'abord nous avons fait le schéma électronique sous Capture CIS d'Orcad 10.5 (voir Fig. 18), puis il a fallu créer les empreintes (PCB footprint) de chaque composant en respectant précisément les datasheets associés dans la bibliothèque (Library Manager) de Layout d'Orcad 10.5 (voir fig. 19).

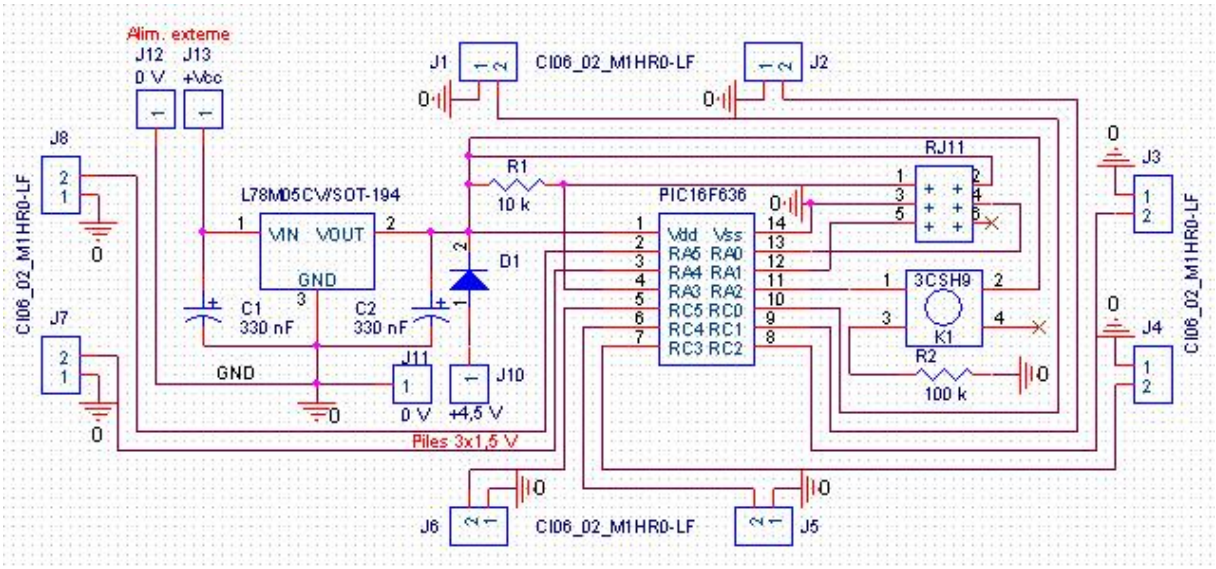


Fig. 18 : Schéma électronique du produit sous Capture CIS d'Orcad 10.5

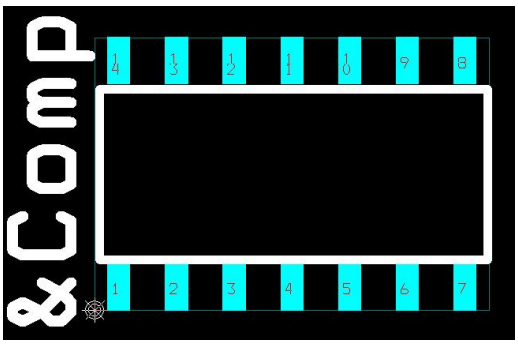


Fig. 19 : Empreinte du PIC 16F636

Ensuite, nous avons réalisé le routage de la carte électronique toujours avec Layout où plusieurs contraintes ont dirigé nos choix. La première était de n'utiliser qu'une seule couche, en l'occurrence le TOP, afin de limiter les coûts de production. Il a fallu donc positionner idéalement les composants. D'une part pour occuper le moins de place possible et rendre le produit plus homogène esthétiquement parlant, et d'autre part pour éviter tout croisement de pistes ce qui a été très difficile. La seconde était d'éviter d'avoir des pistes de faible largeur car évidemment elles se fissurent et rompent plus facilement. C'est pourquoi toutes les pistes font 25 mils de large (100 mils = 2,54 mm et 1 inch = 1000 mils) sauf celles passant entre les pastilles du RJ11 qui font 12 mils et celles passant entre les pattes du PIC qui font 8 mils (voir Fig. 20).

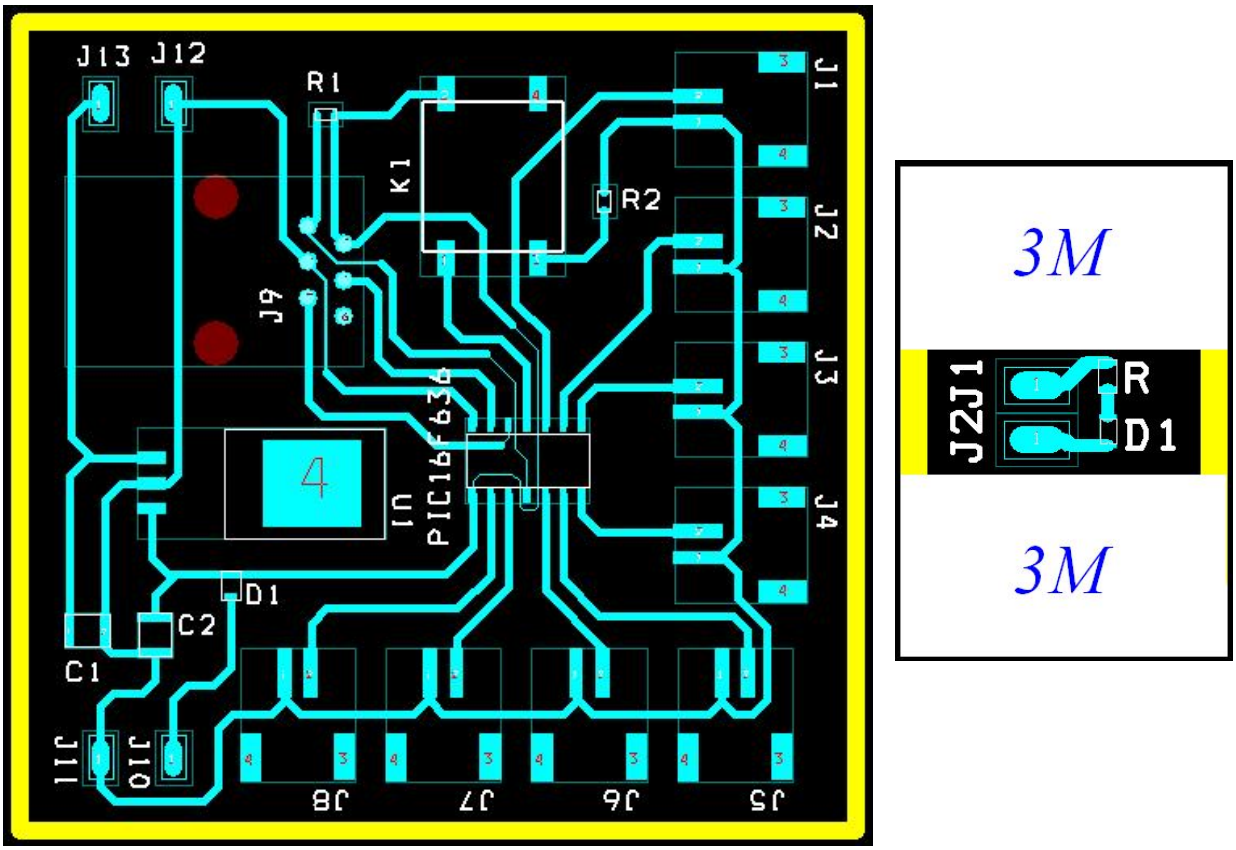


Fig. 20 : Routage de la carte de contrôle et d'une carte à LED avec le scotch double face 3M

Les fichiers GERBER à fournir sont enregistrés comme précédemment dit.

Pour finir, il faut donner les spécifications et préconisations de montage pour éviter d'éventuelles erreurs de compréhension et expliquer l'emplacement des scotchs double face 3M (voir dossier de réalisation). Ainsi, la fabrication automatisée du produit est fin prête et nous devons maintenant rédiger un manuel d'utilisation afin de renseigner au mieux le client sur le produit acheté et rendre son utilisation facile et optimale.

4.3. Rédaction du manuel d'utilisation

Nous ne ferons que décrire la réalisation du manuel puisqu'il est fourni. Nous nous sommes appuyés sur des manuels d'utilisation de matériel hifi afin de reprendre la même organisation et certaines phrases clés. Le manuel est en français mais aussi en anglais.

Tout d'abord, la page de garde contient le logo de l'entreprise TRAFF, le nom du produit, l'intitulée, le choix français/anglais, mais surtout il est important de remercier le client d'avoir acheté le produit, marque de respect, en lui conseillant de lire le manuel pour éviter tout problème matériel ou de montage.

Ainsi, la première étape est la vérification par le client de la présence des cartes et des composants avec l'appui de photos en noir et blanc.

La deuxième étape est la prise de connaissance du système et de ses éléments, puis bien évidemment de ses caractéristiques techniques comme par exemple son autonomie de six semaines. Son fonctionnement est également explicité.

La troisième étape est la mise en service du produit et dans le manuel sont décrites toutes les opérations à effectuer afin de rendre fonctionnel le produit, en particulier est expliquée l'insertion des piles appropriées.

La quatrième étape est la découverte des capacités techniques du produit permettant au client de connaître précisément les animations disponibles.

Enfin, la dernière étape spécifique à notre produit est la fixation des éléments. Elle renseigne le client sur la manière la plus appropriée de profiter au maximum de la flexibilité de fixation du système afin de rendre optimale la mise en valeur du produit à promouvoir.

Au dos du manuel sont données les informations sur l'entreprise TRAFF : l'adresse, le numéro de téléphone et les sites internet où l'on peut découvrir les différents produits commercialisés. Le client a donc toutes les cartes en main pour profiter pleinement de son produit et est rassuré car en plus il peut contacter l'entreprise en cas de problème.

Conclusion

A travers ce projet, nous avons vu toutes les étapes de développement d'un nouveau produit, de la définition du cahier des charges à la finalisation du produit en passant par la réalisation d'une veille technologique, la conception d'un prototype pour valider l'étude théorique et la rédaction du dossier de fabrication.

Au-delà de l'aspect technique, le produit se devait d'être attractif, de taille minimale, facile à fixer et à utiliser. Ceci a donc permis d'approcher le côté marketing du développement d'un produit en répondant au mieux aux besoins de l'acheteur.

Des clients sont déjà intéressés par ce produit, il ne reste donc plus qu'à produire les quantités souhaitées. Ils seront vendus en volume de 200 pièces pour un prix d'environ 8€ pièce, prix auquel Loupi vend une carte munie d'une seule LED sans microcontrôleur.

Notre mission a été menée à son terme et le système sera prochainement lancé sur le marché. Il s'agit d'un produit basique et c'est pour cette raison que l'entreprise TRAFF a demandé à développer le même type de produit à Damien Gerthoffert et Ludmila Slimani, mais avec davantage d'options (réglage de la vitesse d'animation, réglage de l'intensité lumineuse des LED, communication DMX,…) pour répondre à des besoins plus haut de gamme.

Sources documentaires

1. *www.loupi.com*, site de l'entreprise numéro un en application PLV.
2. *www.led-fr.net*, site de renseignement sur la technologie des LED.
3. *www.abcelectronique.com/bigonoff*, site d'aide à la programmation en PIC.
4. *www.radiospares.fr*, site d'un fournisseur de composants électroniques sur lequel les datasheets des composants ont été trouvées.
5. *www.conrad.fr*, site d'un fournisseur de composants électroniques sur lequel les datasheets des composants ont été trouvées.
6. *www.microchip.com*, site du fabricant de microcontrôleurs PIC sur lequel les datasheets ont été récupérées.
7. *www.brilliance-tech.com*, site d'un fabricant de LED haute technologie sur lequel les datasheets ont été récupérées.
8. *www.cvilux.com.tw*, site d'un fabricant de connecteurs CMS sur lequel les datasheets ont été récupérées.

ANNEXES